



# **IMPLEMENTACIÓN DE UN TRANSCEIVER CON PULSOS DE NYQUIST EN FPGA MEDIANTE SIMULINK**

**A Degree Thesis**

**Submitted to the Faculty of the**

**Escola Tècnica d'Enginyeria de Telecomunicació de  
Barcelona**

**Universitat Politècnica de Catalunya**

**by**

**Daniel García García**

**In partial fulfilment**

**of the requirements for the degree in**

**TELECOMMUNICATIONS SYSTEMS ENGINEERING**

**Advisor: Jose Antonio Lázaro Villa**

**Barcelona, January 2017**

## **Abstract**

Nowadays, the high demand of data on communication systems has made essential the optimization of resources, especially in terms of spectral efficiency and responsible use of the network installation.

In this thesis, a software-defined transmitter and receiver will be designed and tested in a real platform. Moreover, a Nyquist pulse will be analyzed and designed to use more efficiently the available bandwidth.

## **Resum**

Actualment, l'alta demanda de dades en els sistemes de comunicacions ha fet essencial l'optimització de recursos, especialment en termes d'eficiència espectral i ús responsable de la xarxa instal·lada.

En aquest treball, es dissenya i es prova un transmissor i un receptor definits mitjançant software. A més, s'analitza i es dissenya un pols de Nyquist per utilitzar més eficientment l'ample de banda disponible.

## **Resumen**

Actualmente, la alta demanda de datos en los sistemas de comunicaciones ha hecho esencial la optimización de recursos, especialmente en términos de eficiencia espectral y uso responsable de la instalación de red.

En este trabajo, se diseña y se prueba un transmisor y un receptor definidos mediante software. Además, se analiza y se diseña un pulso de Nyquist para utilizar más eficientemente el ancho de banda disponible.

## **Agradecimientos**

Me gustaría expresar mi gratitud a Jose Antonio Lázaro, por su dedicación y ayuda constante en este proyecto, pero sobre todo por motivarme siempre, incluso en los momentos de más indecisión. Así también quisiera agradecer a Marc Alonso por ayudarme a empezar y guiarme cuando estaba bloqueado.

Gracias también a esas personas que, sin formar parte del proyecto, me han ayudado a llegar al final.

## Historial de revisión

| Revisión | Fecha      | Propósito              |
|----------|------------|------------------------|
| 0        | 05/12/2016 | Creación del documento |
| 1        | 10/01/2017 | Revisión del documento |
|          |            |                        |
|          |            |                        |
|          |            |                        |

## DOCUMENT DISTRIBUTION LIST

| Nombre                    | e-mail                       |
|---------------------------|------------------------------|
| Daniel García García      | danigarciagarcia94@gmail.com |
| Jose Antonio Lázaro Villa | jose.lazaro@tsc.upc.edu      |

|              |                      |                          |                           |
|--------------|----------------------|--------------------------|---------------------------|
| Escrito por: |                      | Revisado y aprobado por: |                           |
| Fecha        | 05/12/2016           | Fecha                    | 10/01/2017                |
| Nombre       | Daniel García García | Nombre                   | Jose Antonio Lázaro Villa |
| Cargo        | Autor del proyecto   | Cargo                    | Supervisor del proyecto   |

## Índice

|  |    |
|--|----|
| Abstract .....   | 1  |
| Resum .....  | 2  |
| Resumen .....  | 3  |
| Agradecimientos .....  | 4  |
| Historial de revisión .....  | 5  |
| Índice .....   | 6  |
| Lista de figuras .....   | 8  |
| Lista de tablas .....  | 9  |
| 1. Introducción .....  | 10 |
| 1.1. Objetivos .....   | 10 |
| 1.2. Plan de Trabajo .....   | 10 |
| 2. Estado del arte de la tecnología utilizada o aplicada en esta tesis ..... | 12 |
| 2.1. Principios fundamentales de pulsos de Nyquist .....                     | 12 |
| 3. Desarrollo .....  | 13 |
| 3.1. Recursos utilizados .....   | 13 |
| 3.1.1. Software .....  | 13 |
| 3.1.1.1. MATLAB .....  | 13 |
| 3.1.1.2. Simulink .....  | 13 |
| 3.1.1.3. HDL Coder .....   | 13 |
| 3.1.1.4. Optilux .....   | 14 |
| 3.1.1.5. Quartus II .....  | 14 |
| 3.1.2. Hardware .....  | 14 |
| 3.1.2.1. Cyclone V GX Starter Kit .....                                      | 14 |
| 3.1.2.2. THDB-ADA .....  | 15 |
| 3.2. Simulación pulsos de Nyquist con Optilux .....                          | 16 |
| 3.2.1. Potencia transmitida .....  | 17 |
| 3.2.2. Diagrama de ojo .....   | 17 |
| 3.2.3. Bit Error Rate (BER) .....  | 17 |
| 3.2.4. Densidad espectral .....  | 18 |
| 3.3. Diseño del transmisor QPSK .....  | 19 |
| 3.3.1. Data Generation & Packetization .....                                 | 20 |
| 3.3.2. Symbol Mapping .....  | 21 |
| 3.3.3. Pulse Shaping .....   | 21 |

|        |   |    |
|--------|---|----|
| 3.3.4. | Conexión DAC.....                               | 22 |
| 3.4.   | Diseño del receptor QPSK.....                   | 22 |
| 3.4.1. | Conexión ADC.....                               | 24 |
| 3.4.2. | Automatic gain control .....                    | 24 |
| 3.4.3. | Receiver Filter .....                           | 24 |
| 3.4.4. | Fine Frequency Compensation.....                | 24 |
| 3.4.5. | Timing Recovery .....                           | 25 |
| 3.4.6. | Conexión DAC.....                               | 25 |
| 4.     | Resultados .....                                | 26 |
| 4.1.   | Resultados del transmisor .....                 | 26 |
| 4.2.   | Resultados del receptor.....                    | 27 |
| 4.3.   | Resultados en un sistema óptico .....           | 30 |
| 5.     | Coste del proyecto.....                         | 32 |
| 6.     | Conclusiones y futuros desarrollos: .....       | 33 |
|        | Bibliografía .....                              | 34 |
|        | ANEXO A: Características del Amplificador ..... | 35 |
|        | ANEXO B: Características del Láser .....        | 36 |
|        | Glosario .....                                  | 37 |



## **Lista de figuras**

|   |    |
|---|----|
| Figura 1: Plan de trabajo .....   | 11 |
| Figura 2: Diagrama de Gantt .....   | 11 |
| Figura 3 Respuesta impulsional (izquierda) y espectral (derecha) de un pulso de Nyquist ..... | 12 |
| Figura 4 Placa Cyclone V GX .....   | 15 |
| Figura 5 THDB-ADA .....   | 16 |
| Figura 6 Potencia transmitida Optilux .....   | 17 |
| Figura 7 Diagrama de ojo Optilux .....  | 17 |
| Figura 8 BER Optilux .....  | 17 |
| Figura 9 OSNR para BER = $1e-10$ .....  | 18 |
| Figura 10 Densidad espectral Optilux .....  | 18 |
| Figura 11 Diseño transmisor.....  | 19 |
| Figura 12 Data Generation & Packetization .....   | 20 |
| Figura 13 Data Source.....  | 20 |
| Figura 14 Symbol Mapping.....   | 21 |
| Figura 15 Constelación QPSK .....   | 21 |
| Figura 16 Conexión DAC.....   | 22 |
| Figura 17 Conexiones DACs y ADCs con Receptor .....   | 23 |
| Figura 18 Receptor QPSK.....  | 24 |
| Figura 19 Conexión ADC.....   | 24 |
| Figura 20 Pulsos de Nyquist transmitidos .....  | 26 |
| Figura 21 Señal transmitida .....   | 26 |
| Figura 22 Diagrama de ojo de la señal transmitida .....                                       | 27 |
| Figura 23 Espectro de la señal transmitida .....  | 27 |
| Figura 24 Señal transmitida (arriba) y señal recibida (abajo) .....                           | 28 |
| Figura 25 Retardo en la transmisión .....   | 28 |
| Figura 26 Margen dinámico del sistema .....   | 28 |
| Figura 27 Diagrama de ojo de la señal recibida .....  | 29 |
| Figura 28 Espectro de la señal recibida.....  | 29 |
| Figura 29 Estructura sistema óptico.....  | 30 |
| Figura 30 Resultado láser modulado por transmisor.....  | 31 |
| Figura 31 Resultado sistema óptico completo .....   | 31 |



## **Lista de tablas**

|                                  |    |
|----------------------------------|----|
| Tabla 1 Coste del proyecto ..... | 32 |
|----------------------------------|----|

## 1. Introducción

Durante los últimos años, el constante aumento del volumen de datos en Internet ha hecho necesario crear redes de comunicación capaces de soportar el incremento de la demanda de datos. Uno de los problemas de la actual demanda de datos es el uso eficiente de los recursos disponibles, que actualmente limitan la comunicación tanto en velocidad como en capacidad.

Este trabajo se incluye en el proyecto Sustainable Network Infrastructure Enabling the Future Digital Society (SUNSET) del departamento de Teoría del Señal y las Comunicaciones. El proyecto SUNSET tiene el objetivo de superar las limitaciones de las actuales redes de transporte de datos mediante técnicas ópticas avanzadas controladas por Redes Definidas por Software (SDN), mejorando las prestaciones de la red y automatizando el uso responsable e inteligente de los recursos.

Una de las metas principales de SUNSET es la investigación y desarrollo en transmisores y receptores definidos por software. Este proyecto se centra en el desarrollo de un transmisor y un receptor que utilicen pulsos que cumplan el criterio de Nyquist, diseñados mediante el software Simulink de Mathworks, un entorno visual que permite diseñar sistemas por bloques, lo que permite cambiar el sistema de forma más sencilla que el VHDL, el lenguaje de bajo nivel de las FPGAs (Field Programmable Gate Array). Para obtener resultados prácticos se prueban los diseños en dos placas Cyclone V de Altera complementadas con la placa periférica THDB-ADA, idónea para procesamiento digital de señales.

La base del proyecto se apoya en el Trabajo de Fin de Grado de Marc Alonso [1], realizado en el mismo departamento, que consiguió integrar un transmisor diseñado con Simulink en una FPGA, pero no así el receptor. En este trabajo se ha querido estudiar el comportamiento del pulso de Nyquist, diferente al pulso utilizado por Marc Alonso, por lo que el transmisor ha de ser rediseñado para utilizar el pulso conformador deseado y el receptor ha de ser implementado para obtener el sistema completo.

### 1.1. Objetivos

Los principales objetivos del proyecto son:

- Demostrar las ventajas de usar un pulso que cumpla el criterio de Nyquist.
- Diseñar un transmisor y un receptor mediante Simulink de Mathworks.
- Implementar el transmisor y el receptor en dos FPGAs para poder obtener resultados prácticos y analizarlos.

### 1.2. Plan de Trabajo

Para cumplir los objetivos, se han definido las tareas (Figura 1) y el tiempo necesario para completarlas (Figura 2). Estas tareas han sufrido modificaciones respecto al plan original.

Debido a que tanto el diseño del receptor como el del transmisor han sido realizados a partir de modelos dado por Mathworks, la parte de diseño ha consistido en adaptar los modelos a las necesidades del proyecto y el tiempo de estas tareas se ha visto acortado.

Por otra parte, la correcta implementación del diseño del receptor en la FPGA ha sido más costosa de lo previsto inicialmente, por lo que se le ha dedicado más tiempo.

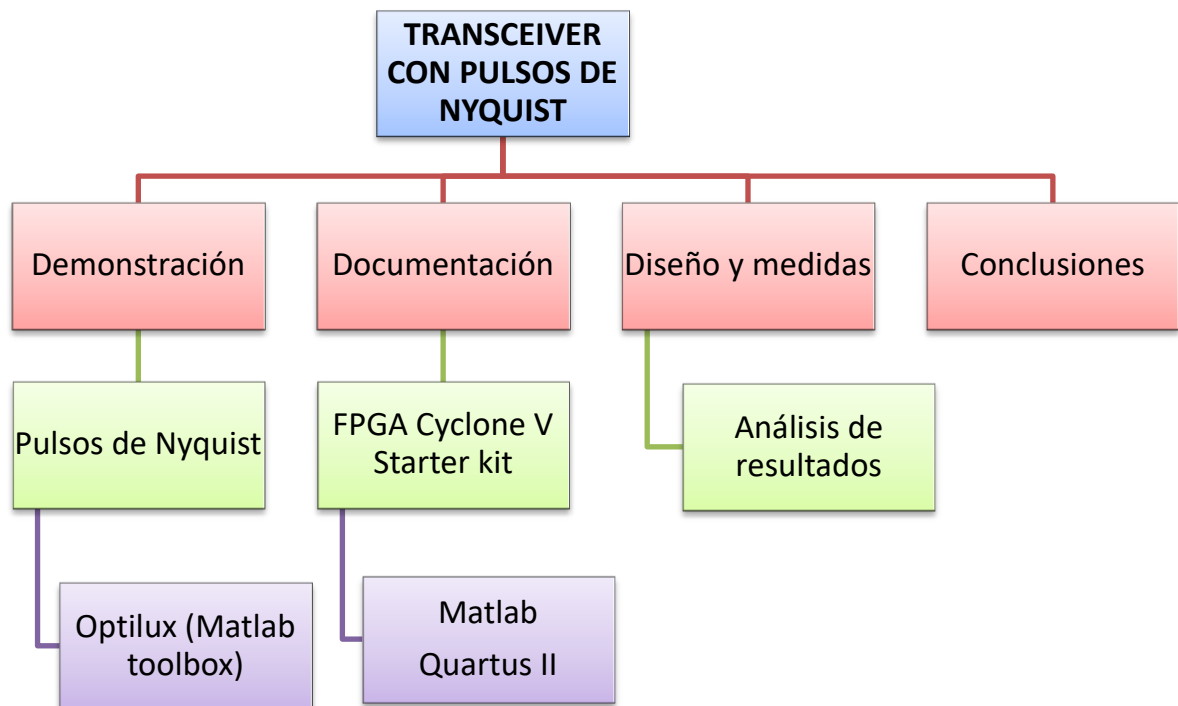


Figura 1 Plan de trabajo

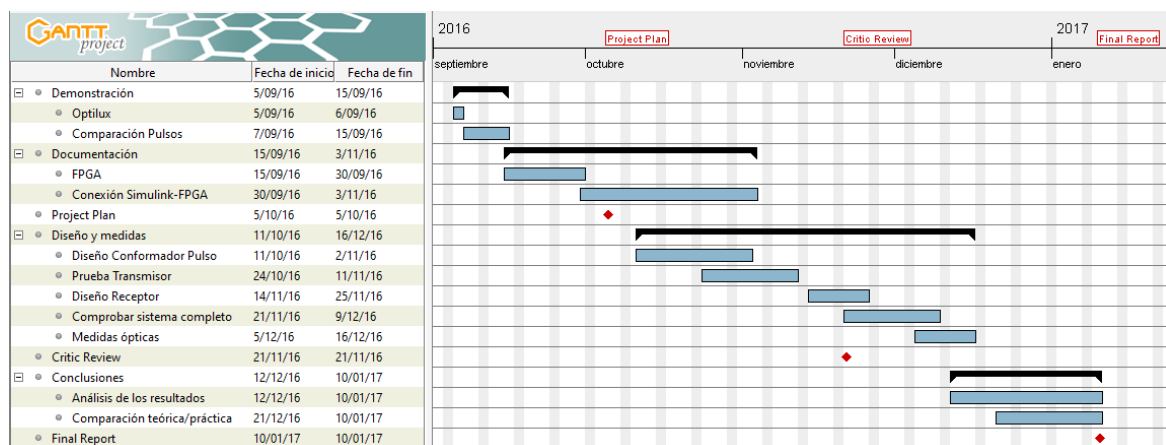


Figura 2 Diagrama de Gantt

## 2. Estado del arte de la tecnología utilizada o aplicada en esta tesis

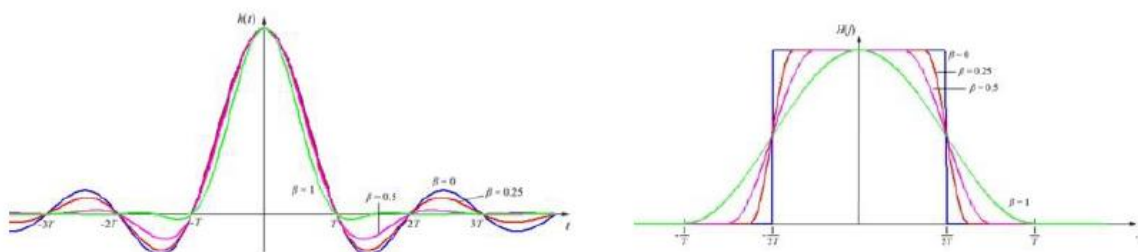
### 2.1. Principios fundamentales de pulsos de Nyquist

En comunicaciones, se conoce como pulso de Nyquist a un pulso que cumple el Criterio de Nyquist para ISI (InterSymbol Interference). Por tanto un pulso de Nyquist no tiene una forma definida, sino que en realidad son todos aquellos pulsos que cumplen ciertas condiciones. El criterio de Nyquist describe las condiciones que ha de cumplir un sistema de comunicaciones para que la señal resultante presente una Interferencia entre símbolos nula.

Una de las características de este pulso es que, aunque su duración exceda la longitud de un solo bit y que, por tanto, los pulsos se solapen, en los instantes de muestreo su contribución en pulsos contiguos es cero. Es decir:

$$s(kT) = \begin{cases} 1 & k = 0 \\ 0 & k \neq 0 \end{cases}; \text{ siendo } k \text{ un entero y } T \text{ el instante de muestreo.}$$

La señal por excelencia que cumple esta condición es la función sinc. Sin embargo, ésta es infinita temporalmente, por lo que no es factible implementarla. Para solventar este problema, se usan aproximaciones de la sinc extendiéndose M símbolos, siendo M un número entero. Este exceso de duración en el eje temporal se traduce en mayor eficiencia espectral.



**Figura 3 Respuesta impulsional (izquierda) y espectral (derecha) de un pulso de Nyquist**

Actualmente, este pulso se utiliza en comunicaciones ópticas porque permite generar señales a gran velocidad y sin presencia de interferencia intersimbólica. Además, ofrece una mayor elasticidad y flexibilidad respecto a la ocupación del ancho de banda [2]. Una de las ventajas en eficiencia espectral de usar este pulso es la eliminación de los intervalos de guarda, que habitualmente son necesarios para evitar interferencias, lo que resulta inevitable en los planteamientos WDM (Wavelength Division Multiplexing).

### **3. Desarrollo**

En este capítulo se explica el desarrollo del proyecto y los recursos utilizados para llevarlo a cabo.

#### **3.1. Recursos utilizados**

Debido a que el proyecto incluye una parte de diseño y una parte de implementación en un dispositivo, las herramientas usadas se han separado en software y hardware.

##### **3.1.1. Software**

Para conseguir el comportamiento deseado en un dispositivo, se ha de generar un archivo de configuración e incluirlo en el hardware, en este caso una FPGA. Los fabricantes de estos aparatos suelen proveer un entorno de desarrollo para generar el software deseado, en este caso se utilizará una FPGA Cyclone V GX de Altera, por lo que el entorno de desarrollo más conveniente es Quartus II, programa provisto por Altera.

Este proyecto ya se podría desarrollar con Quartus II, sin embargo uno de los objetivos del proyecto es crear un sistema que sea fácilmente modificable, y este entorno utiliza como lenguaje de programación VHDL (VHSIC Hardware Description Language), un lenguaje de bajo nivel, y difícil de entender si no se conoce. Por eso, se ha optado por usar Simulink, una herramienta creada por Mathworks y ejecutada desde MATLAB, que permite definir sistemas mediante bloques de manera gráfica.

Además, gracias a la versatilidad de MATLAB para hacer simulaciones, se ha usado esta herramienta con la Toolbox Optilux para simular sistemas ópticos.

##### **3.1.1.1. MATLAB**

MATLAB es una herramienta de software matemático desarrollada por Mathworks, usada para representación de datos, simulación de algoritmos y manipulación de matrices, y muy usada en centros de investigación. Se puede aumentar sus prestaciones gracias a las cajas de herramientas (Toolboxes) que incluye, y otras herramientas que se ejecutan sobre MATLAB como es el caso de Simulink.

##### **3.1.1.2. Simulink**

Simulink [3] es un entorno de programación visual desarrollado por Mathworks y funciona sobre MATLAB. Esta herramienta permite, a partir de bloques predefinidos o creados por el usuario, generar simulaciones de sistemas completos en muchos ámbitos, desde el tratamiento de la señal hasta la ingeniería biomédica.

Debido al uso tan extendido de MATLAB y Simulink, muchas empresas ofrecen compatibilidad con estos entornos, como es el caso de Altera, lo que permite programar una FPGA más claramente que en el idioma nativo de la placa, el VHDL.

##### **3.1.1.3. HDL Coder**

HDL Coder [4] es una Toolbox de MATLAB creada por Mathworks que permite la conversión de un archivo de Matlab o modelo de Simulink a VHDL para hacerlo

entendible para una FPGA, de tal manera que si se quisiera cambiar la placa sobre la que se ejecuta el sistema solo se tendría que cambiar la configuración del HDL Coder sin necesidad de hacer cambios en el sistema.

#### **3.1.1.4. Optilux**

Optilux [5] es una toolbox de MATLAB y Octave desarrollada por Paolo Serena de la Universidad de Parma. Es una colección de herramientas de código abierto que permiten técnicas avanzadas de simulación y análisis de sistemas de comunicaciones ópticas.

En este trabajo ha sido utilizado para analizar las ventajas de un pulso que cumpla la condición de Nyquist respecto otro tipo de pulsos en un sistema óptico.

#### **3.1.1.5. Quartus II**

Aunque se puede programar la FPGA desde Simulink utilizando HDL Coder, es necesario tener instalado Quartus II, el entorno de desarrollo para las FPGA de Altera, ya que MATLAB se integra con Quartus II para sintetizar el código VHDL y descargarlo en la FPGA.

Además, debido a limitaciones de la propia placa, no se puede generar una señal de reloj a la frecuencia deseada desde Simulink o MATLAB, por lo que se ha optado a modificar el código HDL desde Quartus una vez se ha generado el código del sistema.

### **3.1.2. Hardware**

Con el fin de que las señales creadas pueden ser analizadas en un osciloscopio o utilizadas en un sistema óptico, se generan las señales desde una FPGA. Este dispositivo está formado por elementos lógicos programables, además de periféricos para hacer la FPGA más versátil.

Además, incluyen la posibilidad de incluir otros periféricos, como en el caso de este proyecto donde se incluye una placa para mejorar las características de la comunicación.

#### **3.1.2.1. Cyclone V GX Starter Kit**

Para desarrollar el proyecto se ha escogido la placa Cyclone V GX [6] (Figura 4), por ser un dispositivo óptimo por su bajo coste y su capacidad de procesado.

Las características básicas de la Altera Cyclone V son:

- 77K elementos lógicos programables
- 4884 Kb de memoria integrada
- Programación mediante USB Blaster
- 6 transceivers a 3.125G
- Distintos periféricos: LEDs, siete segmentos, HDMI...
- Pines de expansión (generales, Arduino y HSMC)

Esta última característica permite añadir placas hijas que aumenten las capacidades de la FPGA.

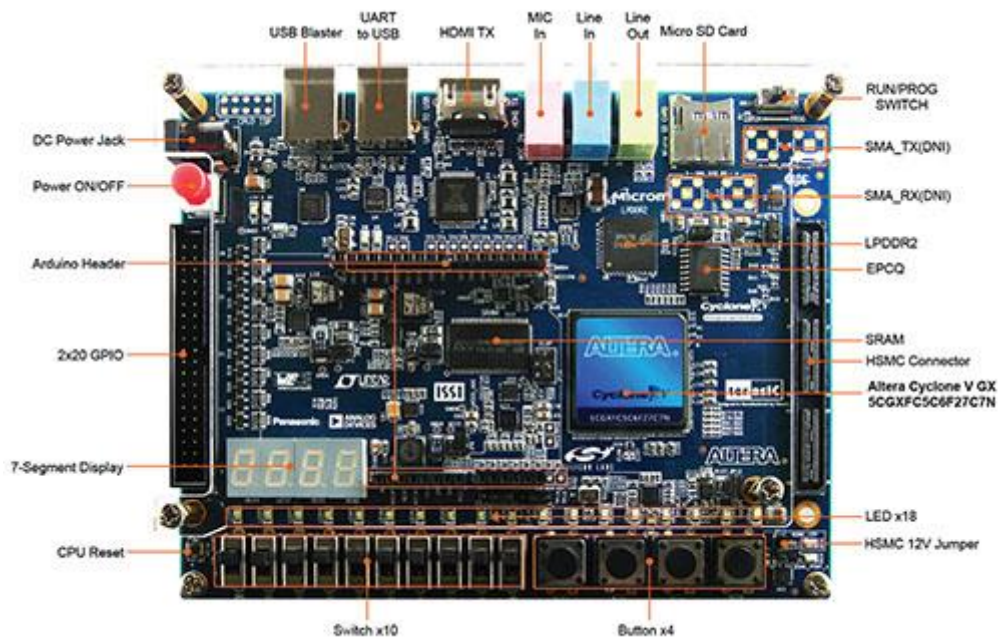


Figura 4 Placa Cyclone V GX

### 3.1.2.2. THDB-ADA

La placa THDB-ADA [7] (Figura 4) de Terasic se puede conectar a la Cyclone V mediante la conexión HSMC y permite aumentar la resolución de la FPGA en las conversiones DA/AD.

Las principales características de la THDB-ADA son:

- Canales duales AD con una resolución de 14 bits y una velocidad máxima de 65 MSPS.
- Canales duales DA con una resolución de 14 bits y una velocidad máxima de 125 MSPS.
- Incluye oscilador interno de 100MHz, SMA para cada AD y DA, y PLL para la interfaz HSMC.
- Entrada con un rango de 2V p-p en el convertidor AD.
- Salida con un rango de 2V p-p en el convertidor DA.
- Tanto el convertidor DA como el AD no admiten señales DC



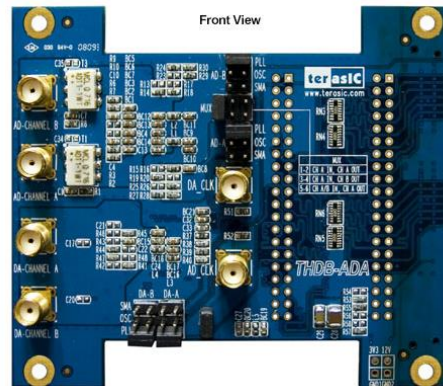


Figura 5 THDB-ADA

### 3.2. Simulación pulsos de Nyquist con Optilux

La primera parte del proyecto consiste en demostrar la mejora al utilizar pulsos de Nyquist respecto otros pulsos en sistemas de comunicaciones ópticas. Para realizar esta tarea se ha utilizado la toolbox Optilux de MATLAB, explicada en el apartado anterior.

Tal y como se explica en el apartado 2, un pulso Nyquist no tiene solo una forma, sino que es todo aquel que cumple ciertas condiciones. Para este proyecto se ha usado una aproximación de una sinc formada por funciones trigonométricas y definida por la siguiente expresión [8]:

**Ecuación 1 Forma de onda del pulso de Nyquist utilizado**

$$p_N(t) = \frac{1}{\sqrt{T}} \frac{\left( \sin\left(\pi \frac{t}{T}(1-\beta)\right) + 4\beta \frac{t}{T} \cos\left(\pi \frac{t}{T}(1+\beta)\right) \right)}{\pi \frac{t}{T} \left(1 - \left(4\beta \frac{t}{T}\right)^2\right)} ; 0 \leq \beta \leq 1$$

Optilux no tiene definido este tipo de pulso por lo que el primer paso es incluirlo con los otros pulsos ya definidos. Además, para reducir el ancho de banda de la señal el pulso se extiende durante M símbolos, siendo M un número entero. Optilux genera la señal uniendo los pulsos necesarios para crear el patrón deseado, por eso se ha cambiado la manera de generar pulsos para que cada pulso se extienda más de un símbolo.

Una vez definido el pulso a analizar se simula el envío de una señal óptica con las siguientes características:

- Número de símbolos: 32
- Número de canales: 1
- Longitud de onda central: 1550 nm
- Baudrate: 10 Gbaud
- Roll-off (del pulso de Nyquist y del coseno alzado): 0.5
- Medidas Back-to-back

Para la simulación se comparan un pulso rectangular, un pulso raíz de coseno alzado y el pulso de Nyquist definido para ver sus diferencias en potencia transmitida, apertura del diagrama de ojo, Bit Error Rate (BER) y densidad espectral.

### 3.2.1. Potencia transmitida

La potencia transmitida por el láser en cada caso es parecida, la que más difiere es el pulso de Nyquist debido a que el pulso se extiende a más de un pulso, por tanto los ceros no son exactamente nulos sino que tienen la aportación de los unos cercanos. Nótese que debido a que la señal se forma de otra manera, hay un pequeño desfase en la señal generada con pulsos de Nyquist.

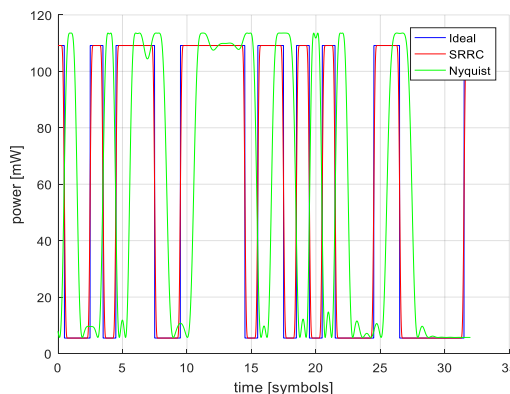


Figura 6 Potencia transmitida Optilux

### 3.2.2. Diagrama de ojo

Los diagramas de ojo tienen aperturas similares, lo que repercute en probabilidades de error similares, que se pueden ver más claramente en el siguiente sub-apartado.

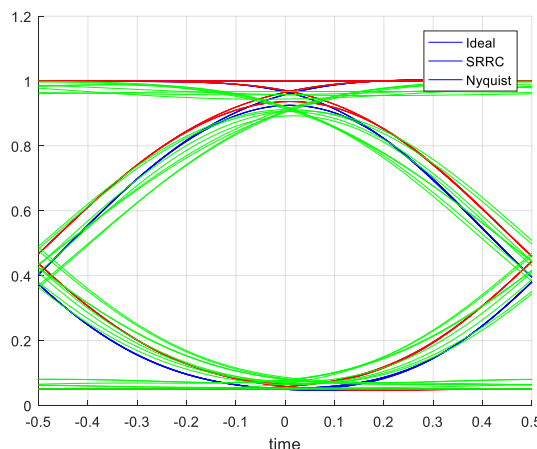


Figura 7 Diagrama de ojo Optilux

### 3.2.3. Bit Error Rate (BER)

En el gráfico adjunto se muestra el Bit Error Rate (BER) cuando la relación señal-ruido óptica (OSNR) en un ancho de banda de 0.1 nm alcanza un determinado nivel. Las curvas demuestran que la probabilidad de error es parecida, siendo la del pulso a analizar un poco peor, es decir, requiere más potencia para tener la misma probabilidad de error. Además se adjunta la OSNR necesaria para tener  $BER = 10^{-10}$ , que confirma la proximidad de las curvas.

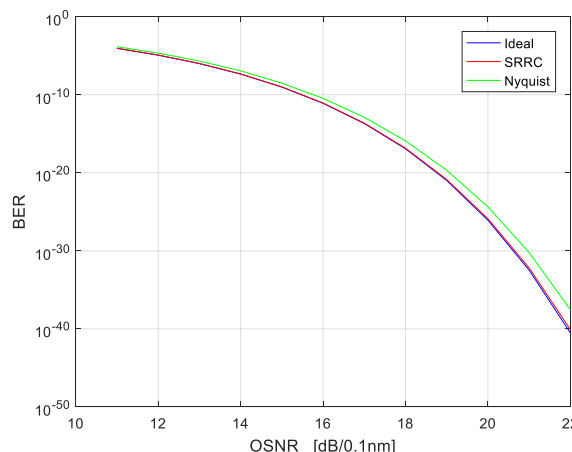


Figura 8 BER Optilux

---

===== Results =====

OSNR B2B (Ideal) @ BER = 1e-10 : 15.4977 [dB/0.1 nm]  
OSNR B2B (SRRC) @ BER = 1e-10 : 15.5111 [dB/0.1 nm]  
OSNR B2B (Nyquist) @ BER = 1e-10 : 15.7809 [dB/0.1 nm]

Figura 9 OSNR para BER = 1e-10

### 3.2.4. Densidad espectral

La densidad espectral de potencia (PSD) es la principal diferencia entre los pulsos estudiados, siendo el pulso de Nyquist el que presenta menor ocupación espectral.

Debido a este aspecto, y siendo similar en el resto de características, este pulso será utilizado en el diseño e implementación del transmisor y del receptor.

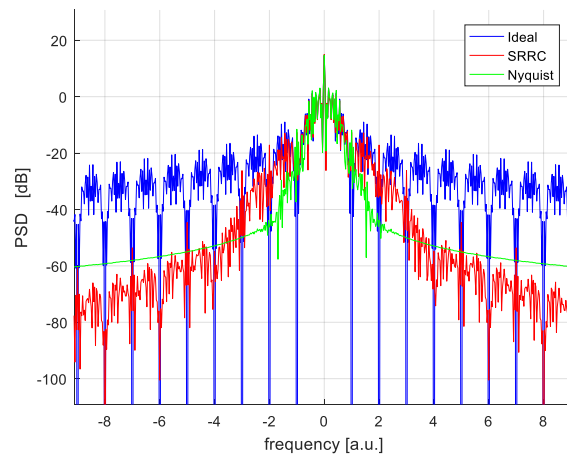


Figura 10 Densidad espectral Optilux

### 3.3. Diseño del transmisor QPSK

En este apartado se explica detalladamente el diseño del transmisor hecho en Simulink y los bloques que lo forman. Los resultados de la señal transmitida corresponden al capítulo siguiente y no se explicarán en este apartado.

Como el principal objetivo de este proyecto es implementar un sistema utilizando un tipo específico de pulso, el formato de modulación a utilizar no es demasiado relevante, por lo que se ha optado por un formato fácil pero que utilice todas las capacidades de la FPGA. Por eso se ha escogido una modulación QPSK (Quadrature Phase Shift Keying), que tiene parte real e imaginaria, y por tanto utiliza los dos canales de la placa hija, pero cada rama solo tiene dos niveles.

Debido a que la meta final es utilizar el diseño en una FPGA, y no todos los bloques de Simulink son compatibles con HDL, se parte de un ejemplo creado por Mathworks de un transmisor QPSK optimizado para HDL [9]. Este modelo se modifica para cumplir los objetivos de este proyecto y se añade un bloque de integración a la placa:

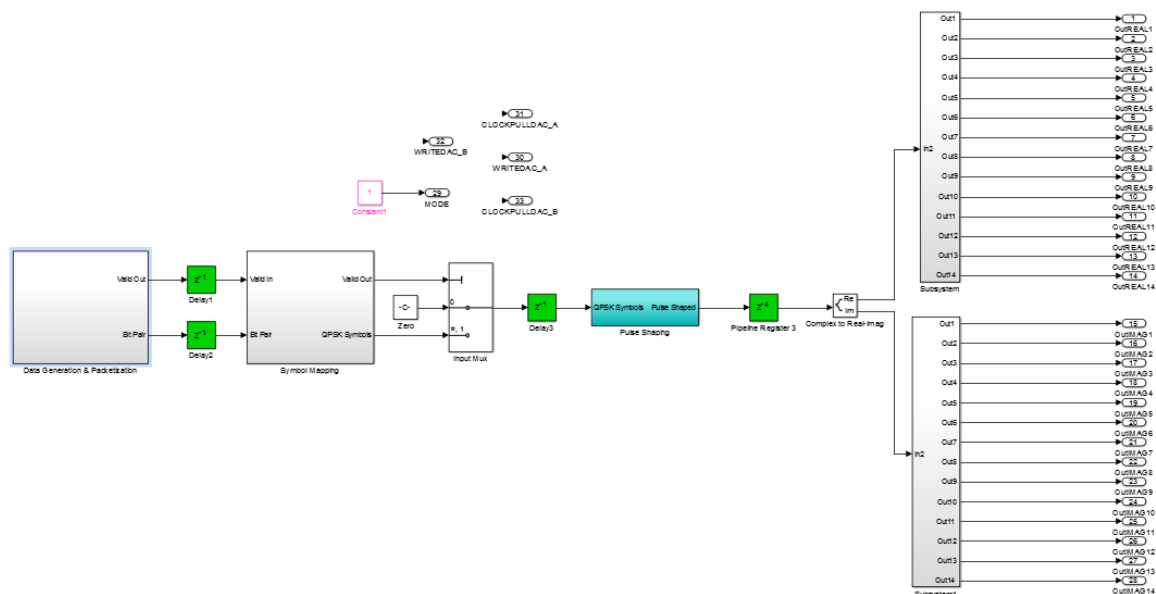


Figura 11 Diseño transmisor

Como se ve en la figura, la estructura la forman principalmente cuatro bloques:

- Data Generation & Packetization: Crea los paquetes a transmitir, agrupando los bits de dos en dos para generar los símbolos.
- Symbol Mapping: A partir de los bits agrupados en el bloque Data Generation & Packetization crea los símbolos QPSK.
- Pulse Shaping: Aplica el pulso de Nyquist diseñado en el apartado anterior y añade muestras.
- Conexión DAC: Adapta a señal para enviarla al convertor DAC (Digital to Analog Converter) de la placa hija añadida a la FPGA.

### 3.3.1. Data Generation & Packetization

El primer bloque es el encargado de crear paquetes y, dentro de cada paquete, agrupar los bits de dos en dos (en el caso QPSK) para crear los símbolos a continuación.

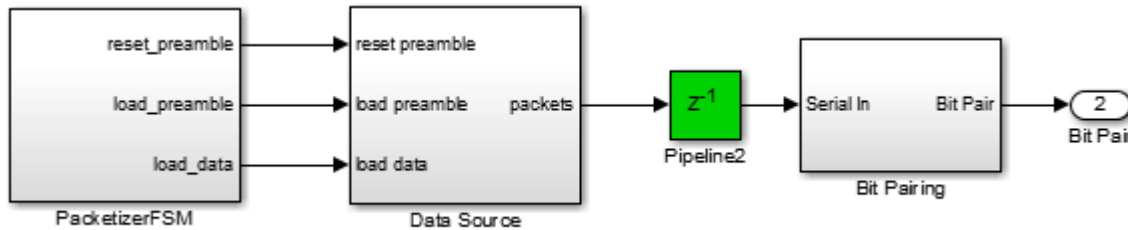


Figura 12 Data Generation & Packetization

Se compone a su vez de tres subsistemas: PacketizerFSM, Data Source y Bit Pairing.

- PacketizerFSM: Es una máquina de estados finita implementada mediante un código MATLAB que tiene dos estados posibles: Pack\_Preamble y Append\_Data. La máquina de estados es básicamente un contador que permanece 26 ciclos de reloj en estado Pack\_Preamble y 174 ciclos de reloj en Append\_Data. Los estados activan o desactivan salidas booleanas para informar al siguiente bloque qué tipo de información ha de añadir.
- Data Source: Este bloque crea los paquetes a partir de la información que le llega del subsistema anterior. Los paquetes están formados por 26 bits de preámbulo de código Barker y 174 bits de datos. Los bits de los que se forman el preámbulo y los datos útiles están almacenados en dos lookup tables.

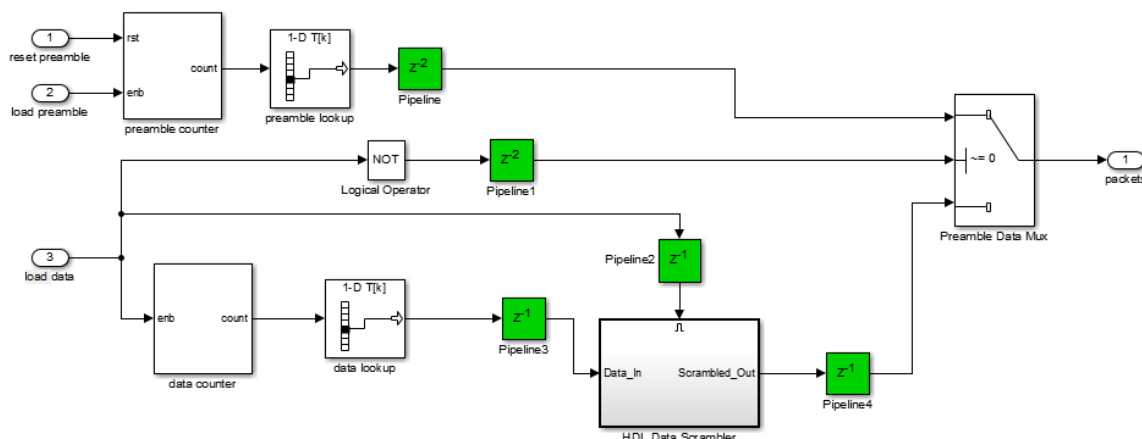


Figura 13 Data Source

- Bit Pairing: Este subsistema agrupa los bits de dos en dos para formar el formato requerido para generar los símbolos.

### 3.3.2. Symbol Mapping

En este subsistema se modulan los bits en banda base para crear la modulación QPSK deseada. Además, se fuerza que la salida sea un entero de 14 bits, el tamaño necesario que requiere la placa hija.

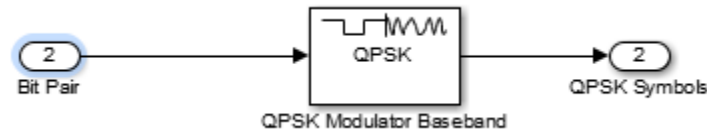


Figura 14 Symbol Mapping

El bloque principal de este subsistema permite escoger parámetros de la modulación. En este caso se ha escogido código Gray, en el que dos valores consecutivos difieren solo en un bit y por tanto se disminuyen los errores, y un desfase de  $\pi/4$ , lo que implica que tanto la parte real como la imaginaria de la modulación solo tiene dos niveles de amplitud, tal y como se observa en la figura siguiente.

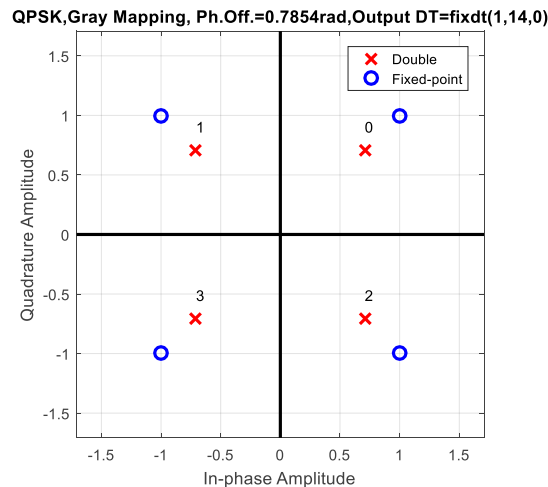


Figura 15 Constelación QPSK

### 3.3.3. Pulse Shaping

Este subsistema está formado por un filtro interpolador con los coeficientes correspondientes a la expresión del pulso escogido en el apartado 3.2 de este trabajo.

El espectro teórico de este pulso, así como la respuesta impulsional, se muestran a continuación:

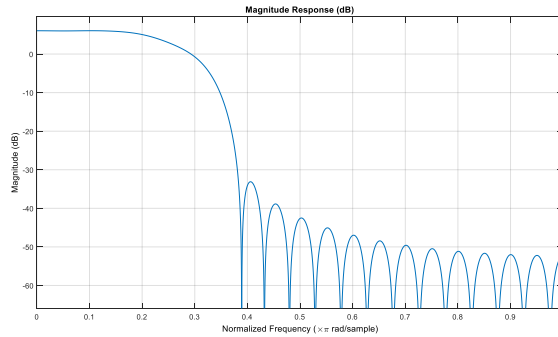


Figura 16 Espectro teórico del pulso

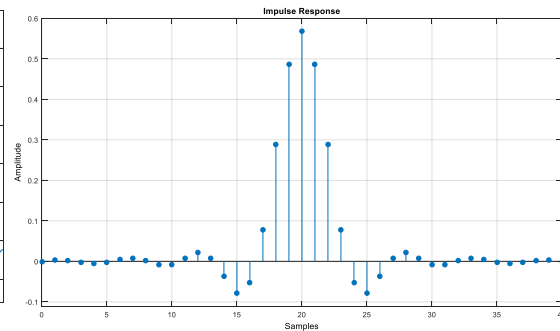


Figura 17 Respuesta impulsional del pulso

### 3.3.4. Conexión DAC

El último bloque permite enviar correctamente la señal a los pines del DAC (Digital to Analog Converter) de la placa THDB-ADA. Es importante que los valores estén en el rango  $[0,1]$ , por lo que la primera parte del subsistema cumple este propósito. Luego se descompone el entero de 14 bits en 14 salidas de un bit para poder asignarlos a los pines de la placa.

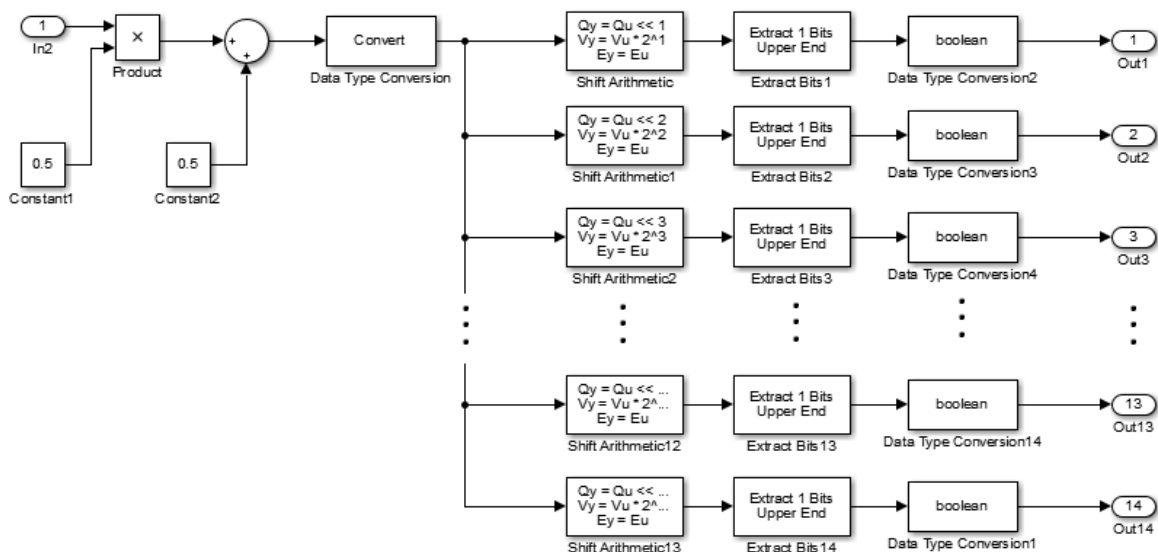


Figura 18 Conexión DAC

### 3.4. Diseño del receptor QPSK

En este apartado se explica detalladamente el diseño del receptor hecho en Simulink y los bloques que lo forman. Los resultados de la señal recibida, así como los resultados del sistema completo, corresponden al capítulo siguiente y no se explicarán en este apartado.

Por limitaciones de la propia FPGA, no se puede monitorizar la señal recibida desde Simulink u otro software del propio ordenador por lo que se ha optado por, una vez la

señal es recibida y procesada por el receptor, volver a sacarla de la placa y visualizarla con un osciloscopio, con el fin de compararla con la señal transmitida (Figura 18).

Como en el caso del transmisor, el diseño no se ha empezado de cero sino que se ha partido de un ejemplo generado por Mathworks de un receptor QPSK optimizado para HDL [10] y debidamente modificado para cumplir los requerimientos de este trabajo (Figura 19).

El receptor diseñado se ocupa de modificar la señal que llega del transmisor para la correcta lectura de datos en tiempo de muestreo. Sin embargo, debido a que no es posible devolver la señal obtenida a Simulink para poder visualizarla como una cadena de bits, el receptor no contiene ningún bloque de muestreo o de decodificación de datos. Por este motivo, el receptor está formado por 4 bloques:

- Automatic Gain Control: Ajusta la señal recibida a un nivel deseado
- Receive Filter: diezma la señal por un factor dos y filtra con el pulso deseado.
- Fine Frequency Compensation: Corrige la frecuencia y la fase.
- Timing Recovery: Sincroniza la señal para poder mostrar en los instantes correctos.

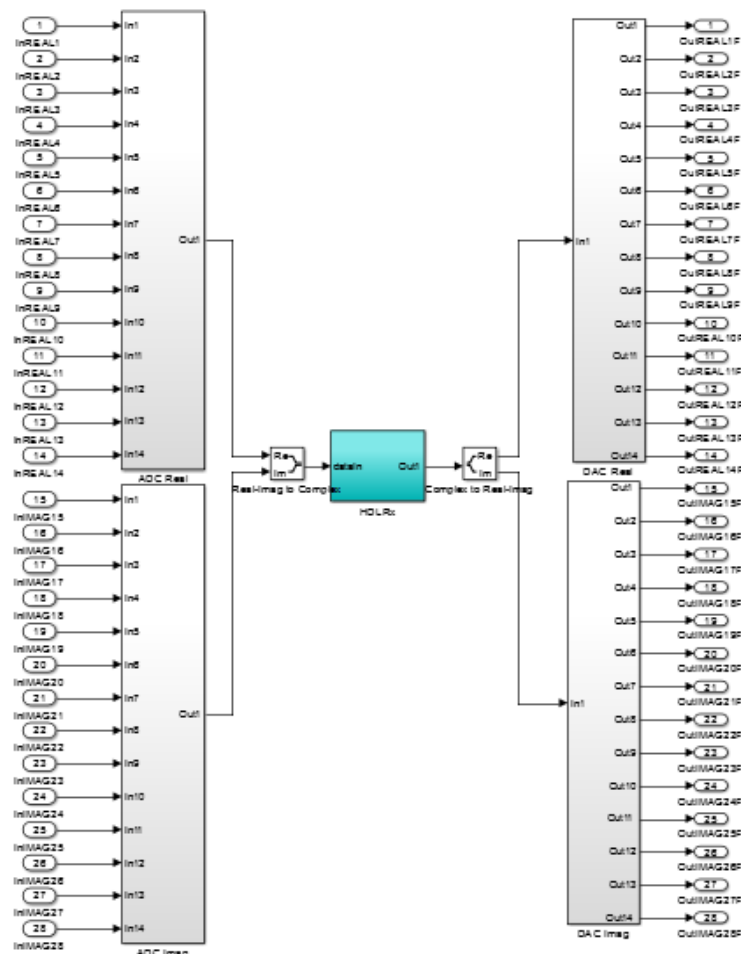


Figura 19 Conexiones DACs y ADCs con Receptor



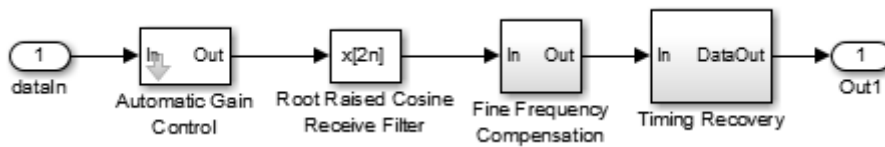


Figura 20 Receptor QPSK

### 3.4.1. Conexión ADC

Cuando la señal llega al ADC (Analog to Digital Converter) de la placa receptora, llega descompuesta en 14 bits la parte real y en 14 la parte imaginaria, por lo que el primer paso es convertir estos bits en un entero, y además que esté en el rango idóneo para hacer cálculos en la FPGA, que es entre -1 y 1.

Par conseguirlo, se concatenan los 14 bits que forman cada rama y se realiza el proceso inverso de la conexión DAC del transmisor: se convierte el entero de 14 bits en un número entre 0 y 1, se resta  $\frac{1}{2}$  y se divide por  $\frac{1}{2}$  para que el valor esté entre -1 y 1. Acto seguido, la señal es tratada en los bloques que son propiamente del receptor.

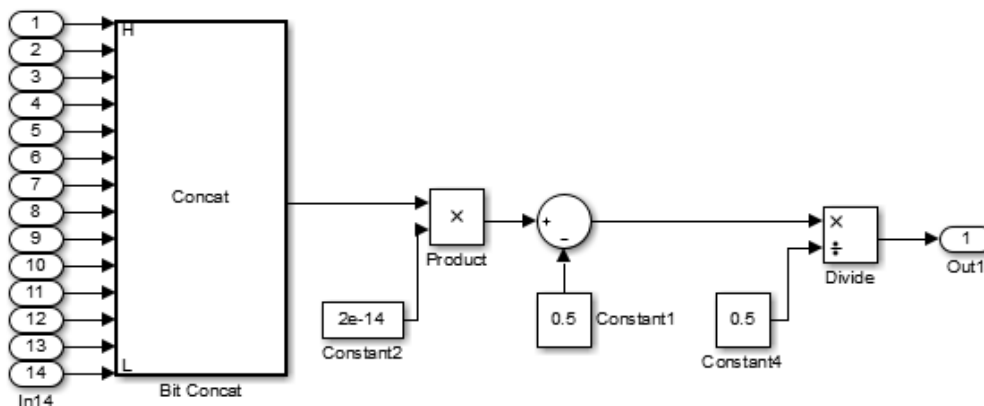


Figura 21 Conexión ADC

### 3.4.2. Automatic gain control

Este bloque se encarga de asegurar una entrada estable para los subsistemas de recuperación de frecuencia y de tiempo. El hecho de colocarlo antes que el filtro receptor mejora la precisión de la estimación, ya que la amplitud se mide con un factor de sobre muestreo de 4.

### 3.4.3. Receiver Filter

El filtro receptor presenta la misma forma que el pulso utilizado en este trabajo, y su finalidad es mejorar la calidad de la señal para conseguir una detección óptima. Además, diezma la señal por un factor dos para aproximar la señal a la secuencia de bits original.

### 3.4.4. Fine Frequency Compensation

El bloque Fine Frequency Compensation usa un Phase-Locked Loop (PLL) para corregir las posibles desviaciones en frecuencia y fase.

#### **3.4.5. Timing Recovery**

El subsistema Timing Recovery utiliza un Phase-Locked Loop (PLL) para corregir los errores de temporización en la señal recibida. De media, el subsistema Timing Recovery genera una muestra de salida para cada dos de entrada.

#### **3.4.6. Conexión DAC**

Este bloque, ya explicado en el apartado 3.3.4, sirve para sacar la señal de la FPGA y visualizarla en un osciloscopio real.

## 4. Resultados

En este apartado se incluyen los resultados de las simulaciones hechas una vez los diseños estaban acabados.

### 4.1. Resultados del transmisor

La primera prueba consiste en comprobar que los pulsos que se transmiten son, efectivamente, de Nyquist. Para ello se conecta una FPGA programada con el diseño del transmisor, y se conecta una de las salidas de la placa THDB-ADA al osciloscopio. En todas las simulaciones de este documento solo se incluyen capturas de la parte real del sistema, aunque todas las conclusiones son aplicables también a la parte imaginaria.

En la imagen del osciloscopio se puede comprobar que, efectivamente, los pulsos son de Nyquist, aunque presentan una forma escalonada debido a la cuantificación de la señal. La señal tiene una amplitud pico a pico de 358 mV y está muestreada a 200 MHz.

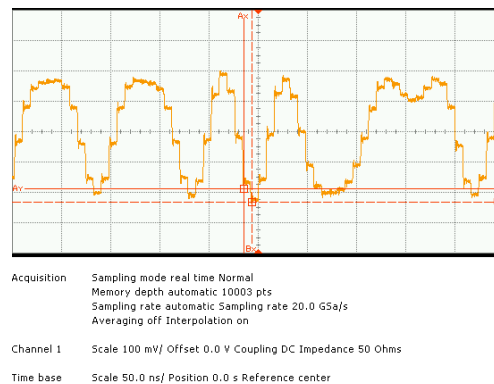


Figura 22 Pulsos de Nyquist transmitidos

Cuando se visualiza un fragmento de señal, se distinguen claramente los dos niveles y el instante donde se debería muestrear cada símbolo, por lo que se concluye que el transmisor cumple su función.

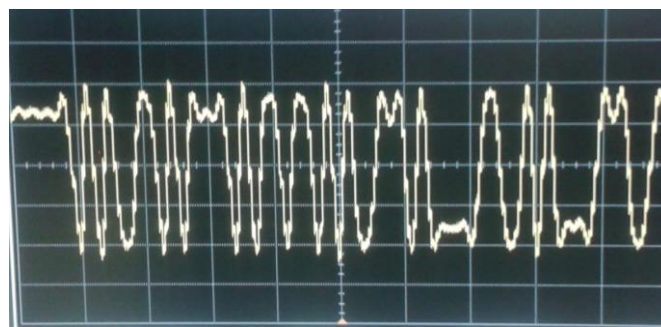


Figura 23 Señal transmitida

Para analizar más a fondo el comportamiento del transmisor se analiza su diagrama de ojo y su espectro.

En el diagrama de ojo se observa la dispersión de los dos niveles, pero están suficientemente separados para evitar errores.

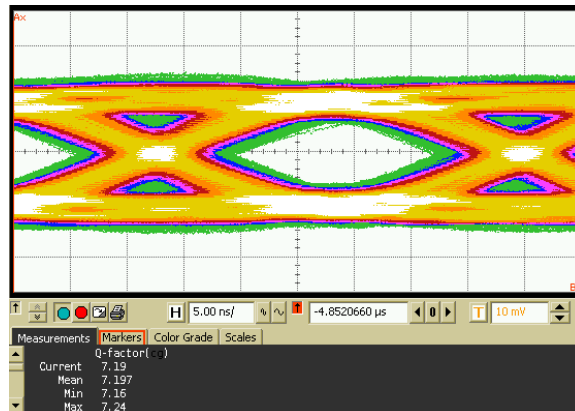


Figura 24 Diagrama de ojo de la señal transmitida

Respecto la densidad espectral, se observa que la señal está modulada en banda base y su comportamiento depende de la respuesta impulsional del filtro conformador. Tal y como se quería demostrar en este trabajo, el pulso de Nyquist es de banda estrecha y decae más de 50 dB en menos de 30 MHz, por lo que mejora el uso de la banda frecuencial respecto un pulso rectangular.



Figura 25 Espectro de la señal transmitida

## 4.2. Resultados del receptor

Una vez se diseña el receptor, se puede probar el sistema completo, conectando una placa que haga de transmisor con otra que haga de receptor. Debido a que no es posible devolver a Simulink la señal recibida, y por tanto no se puede recibir una cadena de bits y decodificar los datos, se comprueba el buen funcionamiento del receptor comparando la señal transmitida con la recibida en un osciloscopio. Además, se comprueban otras características del sistema completo, como el retardo entre el transmisor y el receptor o el margen dinámico.

Al comparar la señal transmitida con la recibida, se comprueba que presentan el mismo patrón, por lo que se concluye que el receptor cumple su propósito. Analizando la amplitud de la señal, se comprueba que el bloque Automatic Gain Control cumple su

cometido, sin embargo podría ser mejorado ya que los picos de la señal presentan una pequeña desviación.

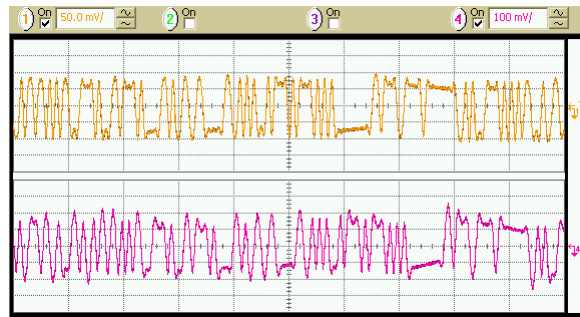


Figura 26 Señal transmitida (arriba) y señal recibida (abajo)

Evidentemente, los cálculos realizados en el receptor retrasan la señal recibida respecto la transmitida, por lo que se mide ese retardo comparando dos muestras iguales de las señales del transmisor y del receptor y se comprueba que el retraso añadido es de 673 ns. En este tiempo también se incluye el retardo ocasionado por la conexión entre el transmisor y el receptor, pero el impacto es mínimo.



Figura 27 Retardo en la transmisión

Para saber el margen dinámico que tiene el sistema se atenúa la señal recibida hasta que los datos útiles se empiezan a confundir con el ruido. En la Figura 28, se muestra el receptor atenuado 120dB, y aunque se sigue distinguiendo la forma de onda, la presencia de ruido podría provocar errores.

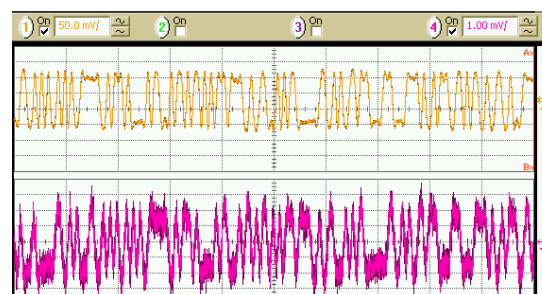


Figura 28 Margen dinámico del sistema (arriba transmisor, abajo receptor)

En el diagrama de ojo del receptor se ve la dispersión de los niveles en el receptor, y es muy parecido al del transmisor, como se puede ver comparando el factor Q del transmisor ( $Q=7.19$ ) con el del receptor ( $Q=7.17$ ). Los errores en el transmisor y en el receptor serán parecidos, aunque bajos debido a que los niveles están suficientemente separados.

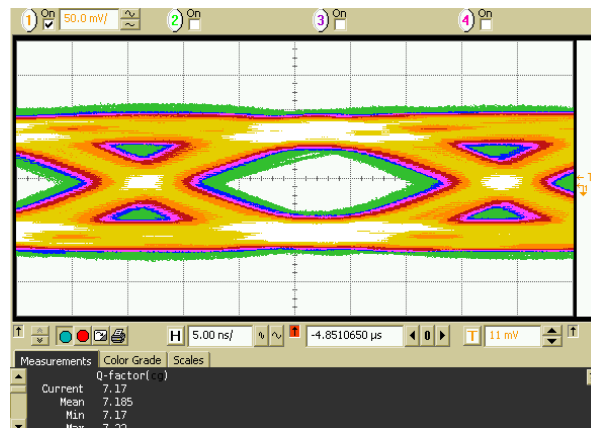


Figura 29 Diagrama de ojo de la señal recibida

Espectralmente, la señal es de banda base y decae más de 50dB a los 30MHz, igual que en el transmisor. Sin embargo, un bloque del receptor, el Fine Frequency Compensation, explicado en el punto 3.4.4, compensa la frecuencia en banda base pero genera dos picos espectrales en los 50 MHz y los 75 MHz, posteriormente eliminados de las medidas mediante un filtro paso bajo en el receptor.

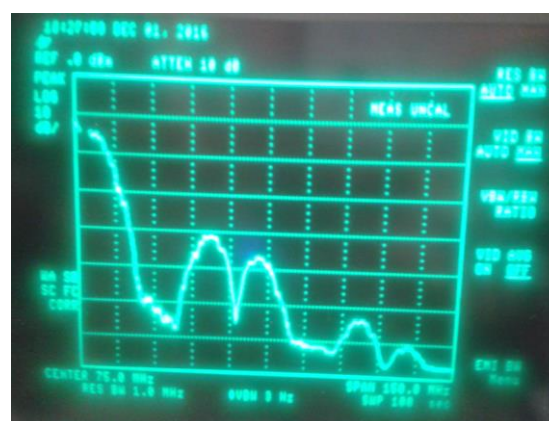


Figura 30 Espectro de la señal recibida

#### 4.3. Resultados en un sistema óptico

Una vez se ha comprobado el correcto funcionamiento del sistema completo, se prueba el sistema añadiendo un láser de onda continua y utilizando el transmisor como modulador externo. El receptor mantiene la misma función conectado previamente a un convertidor óptico a analógico. La estructura de todo el sistema se enseña en la Figura 31 y está compuesta por los siguientes elementos:

1. Una FPGA Cyclone V programada con el diseño del transmisor y haciendo la función de modulador externo.
2. Un amplificador ZFL-1000LN+ de MiniCircuits (ANEXO A), debido a que el circuito de acondicionamiento del láser pide una potencia pico a pico superior de la que provee la placa. Este amplificador no es exactamente de banda base, sino que frecuencia mínima es 100kHz, por lo que corta una pequeña parte de la respuesta frecuencial.
3. El circuito de acondicionamiento de un láser DFB diseñado por Alejandra Díaz-Conti en la Universidad de Zaragoza [11]. El láser que utiliza este circuito es el JDS UNIPHASE CQF935/208 (ANEXO B), que se alimenta a 12 V y su margen lineal es entre 30 y 120mA, siendo ajustado a 80 mA para las pruebas de este trabajo. Este láser incluye un amplificador inversor. Por ese motivo todos los resultados del receptor que se muestran en este apartado se han invertido con el fin de ver la señal con la misma orientación que las envía el transmisor.
4. Una bobina de fibra óptica de 25 km.
5. Un convertidor óptico a analógico para poder leer los datos en el receptor.
6. Una FPGA Cyclone V programada con el diseño del receptor.

Tanto el transmisor como el receptor se conectan al osciloscopio con el fin de obtener resultados gráficos.



Figura 31 Estructura sistema óptico



Antes de probar el sistema completo, se conecta el láser modulado con la FPGA directamente al osciloscopio para comprobar el correcto funcionamiento del láser. En la Figura 32 se confirma que el láser envía la misma información proporcionada por el modulador, con un pequeño overshoot (cuando la señal excede su objetivo) y además la señal está invertida. En la figura ya se ha invertido la señal haciendo uso de las funciones del osciloscopio para que se pueda comparar con la señal transmitida a simple vista.



Figura 32 Resultado láser modulado por transmisor

Al probar el sistema completo, no se puede comparar fácilmente la señal transmitida con la recibida debido al retraso que aporta la bobina de 25 km de fibra, por lo que se analizará la calidad de la señal recibida. Debido a que no hay ninguna fibra ni ningún sistema que compense la dispersión de la fibra, los pulsos han perdido su forma, aunque se siguen distinguiendo los niveles claramente y seguramente se podrían decodificar los datos.



Figura 33 Resultado sistema óptico completo



## 5. Coste del proyecto

| Tarea         | Subtarea                              | Horas      | Coste/hora<br>(€/hora) | Coste Total       |
|---------------|---------------------------------------|------------|------------------------|-------------------|
| DOCUMENTACIÓN | Propuesta del Proyecto                | 10         | 10                     | 100,00 €          |
|               | Revisión crítica del diseño           | 10         | 10                     | 100,00 €          |
|               | Redacción informe final               | 40         | 10                     | 400,00 €          |
|               | Revisión informe final                | 10         | 10                     | 100,00 €          |
| INVESTIGACIÓN | Lectura de tesis y documentos         | 30         | 10                     | 300,00 €          |
|               | Aprendizaje de Simulink               | 10         | 10                     | 100,00 €          |
|               | Aprendizaje de Optilux                | 10         | 10                     | 100,00 €          |
|               | Demostración con Optilux              | 20         | 10                     | 200,00 €          |
| SIMULACIÓN    | Diseño del transmisor                 | 20         | 10                     | 200,00 €          |
|               | Integración del transmisor en la FPGA | 10         | 10                     | 100,00 €          |
|               | Diseño del receptor                   | 40         | 10                     | 400,00 €          |
|               | Integración del receptor en la FPGA   | 50         | 10                     | 500,00 €          |
| MEDICIÓN      | Realizar medidas                      | 70         | 10                     | 700,00 €          |
|               | Extraer conclusiones                  | 30         | 10                     | 300,00 €          |
| HARDWARE      | Cyclone V GX Starter Kit x2           | -          | -                      | 358,00 €          |
|               | THDB-ADA x2                           | -          | -                      | 438,00 €          |
| <b>Total</b>  |                                       | <b>360</b> |                        | <b>4.396,00 €</b> |

Tabla 1 Coste del proyecto

En el cálculo del coste se supone que el proyecto empieza de cero, por lo que el hardware se paga en su totalidad. Además, se ha dado por hecho que se hace el proyecto dentro del marco de la universidad, por lo que no se incluye el coste del software con licencia de estudiante.

## 6. Conclusiones y futuros desarrollos:

Para finalizar el trabajo, se recogen las conclusiones que se han hecho durante el transcurso del proyecto y las posibles líneas futuras que se pueden desarrollar.

En resumen, se ha hecho un estudio comparativo de diferentes pulsos para analizar las ventajas de utilizar un pulso que cumpla el requisito de Nyquist, y acto seguido se ha diseñado un sistema completo con transmisor y receptor que ha confirmado las conclusiones sacadas del estudio comparativo.

Por un lado se quería demostrar la mejora en cuanto a uso espectral de un pulso de Nyquist, como se ha obtenido en las simulaciones con Optilux. Estos resultados se han vuelto a confirmar en las medidas tomadas con las FPGAs programadas con los modelos diseñados en Simulink, que son un primer paso para crear sistemas de comunicaciones en FPGAs sin necesidad de saber VHDL. Además se ha realizado un estudio completo de un sistema con modulación QPSK, dando buenos resultados en un osciloscopio, pero no siendo capaz de decodificar los datos debido a limitaciones de la placa.

Por otro lado, se han utilizado los diseños realizados para modular y recibir en un sistema óptico, con resultados correctos, pero que profundizan poco.

Como posibles futuros desarrollos, se proponen las siguientes tareas:

- Resolver los problemas del reloj y desarrollar un decodificador de datos con la adquisición de la nueva versión de la placa Cyclone V.
- Generalizar los diseños hechos para poder cambiar características del sistema como podrían ser el pulso utilizado, la modulación o la frecuencia de muestreo.
- Ampliar las medidas hechas con el sistema óptico para mejorar las prestaciones del sistema. Por ejemplo, se puede analizar distancias más largas, posibles efectos de dispersión y no linealidad, etc.

## **Bibliografia**

- [1] M. Alonso Varela. "Implementación de un transponder 64-QAM en FPGA mediante Simulink". Degree thesis, Escola Tècnica Superior d'Enginyeria de Telecomunicació de Barcelona, Universitat Politècnica de Catalunya, 2016.
- [2] B Châtelain, C. Laperle. A family of Nyquist pulses for coherent optical communications. *Department of Electrical and Computer Engineering, McGill University, Montreal, Quebec, H3A 2A7, Canada*. 2012
- [3] 2Ciena Corp., Ottawa, Ontario, K2H 8E9, Canada
- [4] Simulink. [Online] Available: <http://es.mathworks.com/products/simulink/>
- [5] Altera Development Board Support from HDL Coder. [Online] Available: <http://es.mathworks.com/hardware-support/altera-development-boards-turnkey.html>
- [6] P. Serena, M. Salsi, M. Bertolini. "Optilux, an Open Source of Light". *Università degli Studi di Palma*, 2009. [Online] Available: <http://optilux.sourceforge.net/>
- [7] Altera, Terasic. "Cyclone V GX Starter Kit" User Manual, 2014.
- [8] Altera, Terasic. "THDB-ADA" User Manual, 2014.
- [9] S. Haykin. *Communication Systems*. 4<sup>th</sup> Edition, New York. Willey, 2001
- [10] Mathworks. HDL Optimized QPSK Transmitter [Online] Available: <https://es.mathworks.com/help/comm/examples/hdl-optimized-qpsk-transmitter.html>
- [11] Mathworks. HDL Optimized QPSK Receiver with Captured Data [Online] Available: <https://es.mathworks.com/help/comm/examples/hdl-optimized-qpsk-receiver-with-captured-data.html>
- [12] A. Díaz-Conti Escribano. "Diseño y Caracterización de un Circuito de Acondicionamiento de un Láser DFB para su Aplicación en Redes Ópticas de Acceso y Metropolitanas". Degree thesis. Escuela de Ingeniería y Arquitectura, Universidad de Zaragoza, 2015.
- [13] Altera. "Black-Boxing in DSP Builder" Application Note 402, October 2005

## ANEXO A: Características del Amplificador

Coaxial

# Low Noise Amplifier

ZFL-1000LN+

50Ω

0.1 to 1000 MHz

### Features

- low noise figure, 2.9 dB typ.
- wideband, 0.1 to 1000 MHz
- protected by US Patent 6,943,629

### Applications

- VHF/UHF
- cellular
- small signal amplifier



Case Style: Y460

| Connectors           | Model       |
|----------------------|-------------|
| SMA                  | ZFL-1000LN+ |
| BRACKET (OPTION "B") |             |

**+RoHS Compliant**

The +Suffix identifies RoHS Compliance. See our web site for RoHS Compliance methodologies and qualifications

### Electrical Specifications

| Parameter                          | Frequency (MHz) | Min. | Typ. | Max. | Units |
|------------------------------------|-----------------|------|------|------|-------|
| Frequency Range                    |                 | 0.1  |      | 1000 | MHz   |
| Noise Figure                       | 0.1-1000        | —    | 2.9  | —    | dB    |
| Gain                               | 0.1-1000        | 20   | —    | —    | dB    |
| Gain Flatness                      | 0.1-1000        | —    | —    | ±0.5 | dB    |
| Output Power at 1dB compression    | 0.1-1000        | —    | +3   | —    | dBm   |
| Output third order intercept point | 0.1-1000        | —    | +14  | —    | dBm   |
| Input VSWR                         | 0.1-1000        | —    | 1.5  | —    | :1    |
| Output VSWR                        | 0.1-1000        | —    | 2.0  | —    | :1    |
| DC Supply Voltage                  |                 | —    | 15   | —    | V     |
| Supply Current                     |                 | —    | —    | 60   | mA    |

## ANEXO B: Características del Láser

### Product Bulletin

#### 2 mW WDM Laser for Direct Modulation in Links up to 100 km CQF915/108 Series

The JDS Uniphase CQF915/108 is a low power, directly modulated laser source in a butterfly package for metro wavelength division multiplexing (WDM) systems up to 100 km.

The laser is pigtailed with a single-mode fiber and shows excellent thermal stability (e.g., wavelength drift with case temperature much better than 1 pm/°C). The package features a high performance thermoelectric cooler for application in uncontrolled ambient temperatures. A cooled isolator minimizes dependence of the optical isolation on case temperature. The internal bias-T network and built-in monitor diode enable simple direct current bias conditioning and output power stabilization of the laser diode.

Under small signal modulation conditions, device bandwidth exceeds 3 GHz. Due to the distributed feedback (DFB) laser's intrinsic low chirp characteristic, the CQF915/108 is especially suitable for digital transmission at 1550 nm. Based on standard single-mode fiber with non-zero dispersion, the maximum dispersion penalty at BER = 10<sup>-10</sup> is less than 1.5 dB for a total dispersion of up to 1800 ps/nm at 1550 nm. JDS Uniphase wavelength drift tests warrant long term wavelength stability.

**Key Features**

- 1550 nm (WDM) DFB laser diode
- 2.5 Gb/s suitable for OC-48 / STM-16
- Built-in high performance thermoelectric cooler
- Built-in cooled optical isolator
- Wavelengths available from 1528 - 1564 nm
- 25Ω electrical impedance matching
- 100 km distance  
(1800 ps/nm dispersion at 1550 nm)

**Applications**

- In WDM metro architectures and in cable television (CATV) networks where low optical output power is required. Especially suitable in environments with uncontrolled ambient temperature.

### Characteristics for All Types ( $R_{th} = 10 \text{ k}\Omega$ , unless otherwise specified)

| Symbol      | Parameter                               | Minimum | Typical | Maximum    |
|-------------|---|---------|---------|------------|
| $I_{th}$    | Threshold current                       |         | 25 mA   | 40 mA      |
| $\lambda_c$ | Central wavelength (ITU grid)           | 1527 nm |         | 1610 nm    |
| $T_k$       | Laser set temperature for $\lambda_c$   | 20 °C   |         | 35 °C      |
| $P_o$       | Output power from pigtail ( $T = T_k$ ) | 10 mW   |         |            |
| SMSR        | Side-mode suppression ratio             | 30 dB   | 45 dB   |            |
| RIN         | Relative intensity noise                |         |         | -135 dB/Hz |

## **Glosario**

|               |  |
|---------------|--|
| <b>ADC</b>    | Analog to Digital Converter  |
| <b>BER</b>    | Bit Error Rate   |
| <b>DAC</b>    | Digital to Analog Converter  |
| <b>DFB</b>    | Distributed FeedBack   |
| <b>FPGA</b>   | Field Programmable Gate Array  |
| <b>HDL</b>    | Hardware Description Language  |
| <b>ISI</b>    | InterSymbol Interference   |
| <b>OSNR</b>   | Optical Signal to Noise Ratio  |
| <b>PLL</b>    | Phase-Locked Loop  |
| <b>PSD</b>    | Power Spectral Density   |
| <b>QPSK</b>   | Quadrature Phase Shift Keying  |
| <b>SUNSET</b> | Sustainable Network Infrastructure Enabling the Future Digital Society |
| <b>WDM</b>    | Wavelength Division Multiplexing                                       |